Doc Code: AP.PRE.REQ

PTO/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW	Docket Number (Optional) ORA005 US

CERTIFICATE OF FACSIMILE TRANSMISSION
I hereby certify that this correspondence is being facsimile transmitted to the U.S. Patent and Trademark Office to the fax number 571-273-8300

on 4/7/2006.

Signature S. Omkar

Typed or printed name OMKAR K. SURYADEVARA

Application Number 09/945,118	Filed August 31, 2001
First Named Inventor Raj Kumar	
Art Unit 2192	Examiner Chameli Das

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

□ applicant/inventor.

□ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

☒ attorney or agent of record.
Registration number 36,320

□ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 ______

S. Omkar
Signature

OMKAR K. SURYADEVARA
Typed or printed name

408-982-8203
Telephone number

4/7/2006
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

□ *Total of ______ forms are submitted.

REASONS FOR REQUESTING PRE-APPEAL BRIEF REVIEW OF US «09/945,118»

All claims stand rejected over Burke's patent either alone or in combination with another reference. However, the only thing that appears to be common between Applicants' claims and Burke's patent appears to be the presence of some words, which words are used in different contexts, with different meanings, and are interrelated to one another in different ways. Burke when properly interpreted by a skilled artisan does not support the rejection of Applicants claims.

Claim 1 is not anticipated by US 6,789,252 granted to Burke for at least eight reasons.

A ***first*** distinction over Burke's patent is that an instance of an application as recited in Claim 1 is not the same as (and is therefore not disclosed or suggested by) an instance of an object in object oriented programming (OOP). For examples of application instances, see Applicants' originally-filed specification, at page 1 lines 23-26 which describes "multiple instances (e.g. Instance1 and Instance 2 in FIG. 1) of the database server". Also see the specification at page 2 at lines 21-23 which states that "a single instance of a database process (also called "Oracle instance") is executing on each of the computers ..." Claim 1's instance is explicitly described in the claim itself as follows: "wherein each instance of the application comprises a plurality of processes." This limitation of Claim 1 was rejected for anticipation by Burke's column 6 lines 10-30 and column 5, lines 30-59. Nothing in Burke's text discloses an instance of an application.

Burke's object instance is created by object-oriented-programming (OOP) cloning. In Burke's patent, a typical object instance is an instance of a datatype (class), such as the Java class "Integer" e.g. Integer i, j; wherein i and j are instances of the class (type) Integer. Burke explicitly states that "[i]n a preferred embodiment, the invention is implemented in the Java programming language, relying substantially on its ***object-oriented programming*** techniques" (emphasis added; see Burke's col. 30 lines 50-52).

Burke fails to disclose an application instance which contains multiple processes as recited at the end of Claim 1. Specifically, a multi-process application is nowhere disclosed in the entirety of Burke's patent. Moreover, Burke's patent fails to disclose multiple instances, of such a multi-process application. The term "object" as recited in Claim 1 is used in its general sense. Claim 1's object has nothing to do with object oriented programming (i.e. Claim 1 does not require nor does it preclude use of an object inheritance mechanism). This interpretation of the term "object" is apparent to a skilled artisan from the claim language and the specification, e.g. at page 6 lines 2-11.

A ***second*** distinction over Burke's patent is that Claim 1 does not explicitly recite creation of a new instance (i.e. a new set of processes) of an application. Instead, Claim 1 recites that a new object is created, from an existing object that is used by an existing instance (i.e. an existing set of processes) of the application. Nothing in the above-quoted text from Burke's patent discloses creating a new object outside of the context of a process. As noted above, Burke's patent at most discloses creation of object instances, by OOP cloning, which results in objects within the context of the process (which created the objects), because the cloned object inherits the source object's model parentage (see Burke's column 20 line 32). Hence, Burke fails to disclose creating a new object outside of process context, which is required if an existing application instance's existing object is used to create a new object for a new application instance, as recited in Claim 1. Specifically, Claim 1's creating is performed outside of the new application (which is to be started), a concept nowhere disclosed by Burke.

Claim 1 requires that after creation of the new object, this newly-created object is used (by the new application instance) to access a resource that is shared by multiple instances (i.e. multiple sets of processes) of the application. The multiple instances in Claim 1 include the existing instance whose existing object has been used to create the new object. Hence a ***third*** distinction of Claim 1 over Burke's patent is that Burke fails to disclose a resource shared by multiple instances of the multi-process application.

A ***fourth*** distinction of Claim 1over Burke's patent is that there is no disclosure or suggestion (in the above-quoted text from Burke's patent) for the shared resource to be accessed by the new application instance using the newly created object. This limitation in Claim 1 was also not addressed in the Final Office Action. This is an additional reason why the Final Office Action fails to make a prima facie rejection, and must be withdrawn.

In view of the above four arguments, Applicants respectfully submit that Claim 1's "creating" limitation (when interpreted as described in the wherein clauses thereof) is nowhere disclosed by the Burke patent. Although to overcome an anticipation rejection, it is sufficient to show that a single claimed limitation is not found in the cited reference, Applicants respectfully submit that Claim 1's remaining limitations are also not disclosed by Burke, as discussed next.

The Examiner states that Claim 1's "setting up a connectivity between the new instance and the network" is disclosed in Burke's patent at column 25, lines 6-14 and column 26 lines 1-16. However, nothing in the cited text even remotely suggests the setting up of connectivity between an instance of a multi-process application, and a network. Instead, all of this text by Burke at most suggests use of pre-existing network connections to support interactions between suppliers and customers using business object definition components. As will be apparent to the skilled artisan, suppliers and customers are two specific entities, and therefore Burke merely teaches their interactions, and permits a user to view an object's properties. However, the claimed setting up of connectivity between the new application instance and a network is nowhere disclosed by Burke. Hence, this is a ***fifth*** distinction of Claim 1 over Burke's patent.

The Examiner states that Claim 1's "starting execution of a new instance of the application" is disclosed in Burke's patent at column 25 lines 38-50, column 56 lines 62-65, and column 4 lines 37-42. Nothing in the cited text discloses the start up of a multi-process application instance. Instead, Burke at most suggests that one or more objects may be launched, resulting in execution. As would be apparent to the skilled artisan, the startup of multiple processes of an application instance is not disclosed merely by the execution of an OOP object. This is a ***sixth*** distinction of Claim 1 over Burke's patent.

Moreover, although several of the words in Claim 1 are found in Burke's patent, Applicants respectfully submit that Burke's words are used in a different context and with a different meaning than the words in Claim 1. As one example, Burke's instances are instantiations of OOP objects, whereas Claim 1's instances are instantiations of a multi-process application. As another example, Burke's applications are business applications (such as a product composition system, an order management system, and a customer ordering system), whereas Claim 1's application can be any application in a computer that comprises multiple processes. Therefore, this is a ***seventh*** argument for patentability of Claim 1 over Burke's patent.

Even if Burke's words were to be used in the same context as and with the same meaning as Claim 1, Burke's steps have different interrelationships among each other and result in a different combination than the method recited in Claim 1. As stated by the Court of Appeals for the Federal Circuit, "An anticipating reference must describe the patented subject matter with sufficient clarity and detail to establish that the subject matter existed and that its existence was recognized by persons of ordinary skill in the field of the invention." See ATD CORPORATION v/s LYDALL, INC. (Fed. Circ. 1998), citing In re Spada, 911 F.2d 705, 708, 15 USPQ2d 1655, 1657 (Fed. Cir. 1990) and Diversitech Corp. v. Century Steps, Inc., 850 F.2d 1566, 1567, 7 USPQ2d 1315, 1317

(Fed. Cir. 1988). Burke's patent simply fails to disclose the identical method recited in Claim 1. This is an ***eighth*** distinction of Claim 1 over Burke's patent.

Claim 2 was rejected as being anticipated by Burke, but Burke fails to disclose making a copy of an existing object, to rename the copy as per Claim 2. The Examiner-cited text at column 20 lines 21-26 and column 54 at line 9 in Burke's patent fails to anticipate Claim 2 for at least two reasons. First, the clone being made by Burke appears to be an instance of an object which is accessed using its "new identifier" (column 20 line 23). Burke's "new identifier" appears to be a reference to a memory location, which memory reference is typically local to the process context, and is therefore not available for use by another application instance as claimed. Second, Burke's use of a "new identifier" teaches away from changing the name of an object. While Burke does disclose that properties of an object may be edited, Burke fails to disclose that the object itself is to be renamed. Therefore, there is no disclosure to use the name of a multi-process application while renaming the copied object, as per Claim 2.

Also, Claim 14 was rejected over the disclosure in Burke at column 25 lines 6-25. However, nothing in the cited text discloses a map file that is shared across all computers, as recited in Claim 14. Moreover, nothing in this text discloses the addition of an entry for the new instance in such a map file, that identifies which instances are running on which computers (e.g. which application instance is running on which computer).

Claim 29 was rejected over the teachings of Burke's column 33 lines 50-60 and column 13 lines 55-60. Burke's column 33 merely teaches object-to-relational mapping, i.e. transformation between OOP and relational models of computer programming. And Burke's column 13 teaches protocol mapping, i.e. conversion of a message that conforms to one protocol into another protocol. Applicants respectfully submit that both of the above-quoted teachings by Burke fail to disclose a map file that identifies a mapping between application instances and computers in a cluster (as noted above, which application instance is running on which computer in the cluster).

Note that the above-described rejection of Claim 29 is symptomatic of a problem with all claim rejections. Specifically, the only thing common among Columns 33 and 13 of Burke and Claim 29 is that all of them use the word "map". However, this word "map" is used in three different meanings, in column 33, column 13, and Claim 29.

Applicants respectfully submit that the mere presence of the same words in Burke's patent is insufficient to support the anticipation rejection of Applicants' claims.

The Examiner also rejected several claims, for being obvious over certain combinations and Applicants will now discuss some of these rejections.

Claim 10 was rejected over the teachings of Feuerman's patent in column 4, lines 57-62. Nothing in the cited text discloses any checking, prior to installation of the data delivery program. Moreover, an **explicit limitation** in Claim 10 "if said second computer does not have said software" is nowhere disclosed by Feuerman (nor is it disclosed in Burke). This is a **prima facie** defect.

Claim 27 was rejected over the teachings of Snyder's patent in column 14, lines 13-24. Note that Snyder teaches storing the persistent data in the "refdata" of his OOP object. Therefore, Snyder fails to disclose or suggest that the persistent data is for a multi-process instance of an application, i.e. Snyder fails to cure Burke's defect as per the first argument above. Moreover, Snyder fails to disclose or suggest that his persistent data is to be stored in a resource that is shared by multiple instances of an application. The Examiner has not explained why a Configuration System which includes all attributes cannot be used as described by Burke at column 47, line 16. Furthermore, adding Snyder's Creation functions to Burke's patent will cause confusion because Burke already discloses a "Create function" (at column 20, lines 14-18). Therefore, Snyder's Creation functions if replacing or adding to Burke's system would require a re-design, thereby teaching away from the Examiner-proposed combination.

Also, Claim 7 requires displaying a list and receiving a selection from the list which is nowhere disclosed by Burke's patent, as admitted by the Examiner. In fact Burke's patent teaches away from using Claim 7's list, by disclosing a Network Editor for graphically creating and maintaining a network definition (see Burke's column 25 at lines 19-25). The Examiner has not explained why Burke's Network Editor does not already provide a single screen and user-friendly environment which is cited as the motivation - in fact Burke does, as shown in FIG. 40A and 40B in Burke's patent. Hence the motivation is already satisfied by Burke's graphical interface and there is no reason to replace it with a list. Adding Feuerman's list to Burke's patent requires Burke's patent to be modified to support the "subscriber" feature of Feuerman's patent. Such a modification requires a re-design of Burke's system, thereby teaching away from the Examiner-proposed combination. Moreover, as per the proposed motivation, Feuerman's list can be added to any prior art reference whatsoever if it contains multiple "elements". This proposed motivation is so generic and overbroad that it is insufficient to support an obviousness rejection.